



高等学校教材 · 计算机系列

Computer
Architecture



计算机系统结构 (第二版)

尹朝庆 主编

华中科技大学出版社
<http://press.hust.edu.cn>



高等学校教材 · 计算机系列

TP303

134

2006

Computer
Architecture

计算机系统结构

(第二版)

主编	尹朝庆		
编著	尹朝庆	陈	利
	夏又新	王	莹

华中科技大学出版社

图书在版编目(CIP)数据

计算机系统结构(第二版)/尹朝庆 主编
武汉:华中科技大学出版社,2006年8月
ISBN 7-5609-2142-6

I. 计…
II. 尹…
III. 计算机体系结构-高等学校-教材
IV. TP303

计算机系统结构(第二版)

尹朝庆 主编

责任编辑:叶见欣
责任校对:陈 骏

封面设计:潘 群
责任监印:张正林

出版发行:华中科技大学出版社
武昌喻家山 邮编:430074 电话:(027)87557437

录 排:华中科技大学惠友文印中心
印 刷:湖北新华印务有限公司

开本:787×1092 1/16	印张:16.75	字数:387 000
版次:2006年8月第2版	印次:2006年8月第5次印刷	定价:25.80元
ISBN 7-5609-2142-6/TP·366		

(本书若有印装质量问题,请向出版社发行部调换)

内 容 简 介

本书以提高计算机性能的并行化概念、方法和技术为主线,以计算机性能的分析计算方法为依托,介绍计算机系统结构的基本概念和基本方法、提高计算机性能的关键技术和性能分析方法。全书共7章,第1章介绍计算机系统结构的基础知识;第2章讨论与指令系统有关的数据表示和指令格式优化设计方法;第3章在介绍流水技术的基础上,着重讨论单发射与各种多发射流水处理机的关键技术以及性能分析方法;第4章介绍存储系统的构成方法和关键技术以及性能分析方法;第5章讨论高性能计算机系统中的互连网络;第6章介绍单指令流多数据流计算机系统;第7章介绍当代主流的各类高性能计算机的结构、特点、关键技术以及并行计算的性能分析。

本书内容详实,层次清晰,详略适当,重点突出,语言严谨精炼,便于教学。可作为高等院校计算机等相关专业的教材,也可作为有关专业的教师和科技工作者的参考书。

前 言

随着科学技术和社会文明的发展,计算机应用领域对计算机性能的要求越来越高。高性能计算机不仅极大地促进社会经济和科技的发展,也成为社会进步和国防安全的重要工具。高性能计算机的应用已从科学研究、工程计算、虚拟试验等领域发展到更多的应用领域,在社会信息化建设中发挥十分重要的作用。高性能计算机的研制与应用一直是计算机科学技术的主要发展方向。

计算机系统结构(Computer Architecture)是高等学校计算机类专业的一门专业核心课程。该课程以高性能计算机为目标,由浅入深地讲授计算机结构设计的基本概念、采用的方法和技术。由于提高计算机性能的主要推动力是提高计算机体系结构的并行性,因此,该课程应以计算机系统并行化的概念、方法和技术为主线,以性能分析方法为依托组织教学内容,并将该领域不断出现的新概念、新方法和新技术适当地补充到教学内容中。

本书的第一版由华中科技大学出版社出版后,被许多高等学校使用,得到广大读者的肯定。根据近几年计算机体系结构的发展趋势和国内外高等学校该课程教学内容的重点,作者重新编写了本书。与第一版比较,第二版的主要改变有:

1. 增加了对当代主流的高性能计算机的结构和关键技术的介绍。
2. 在内容编排上,进一步突出计算机性能分析对学习体系结构设计方法与技术的重要作用与两者之间的关系。
3. 在引入新概念和新技术的同时,删减和更新比较陈旧的内容,全书结构更紧凑,内容更精炼。
4. 各章都精选了一批例题和习题,帮助读者易于理解讲授的概念、技术和方法。

本书共分7章,各章内容如下:

第1章计算机系统结构导论,介绍计算机体系结构的基本概念、基本性能指标和高性能计算机的发展。

第2章指令系统,介绍数据表示和指令格式优化设计方法,以及精简指令集计算机的概念。

第3章流水技术与流水处理机,讨论流水处理机的分类、调度和相关处理等关键技术以及性能分析方法;介绍几种多发射处理机的结构与性能比较;介绍向量流水处理机的结构和提高性能的方法。

第4章存储系统,介绍存储系统的层次结构与性能指标;介绍两种并行存储器的结构;讨论虚拟存储器和高速缓冲存储器的地址变换和替换算法及其实现技术以及对存储系统性能的影响。

第5章互连网络,介绍规则定制的互连网络表示互连的互连函数,以及互连网络的结构参数和性能指标;讨论几种静态互连网络和动态互连网络的结构和性能;讨论由互连网络互连的

处理结点之间的消息传递方式。

第6章单指令流多数据流计算机,介绍单指令流多数据流计算机的两种基本结构及其两种典型的机器实例;讨论阵列处理机的常用算法及其性能分析。

第7章多指令流多数据流计算机,在介绍多指令流多数据流计算机的结构模型和访存模型的基础上,主要围绕高性能计算机体系结构,讨论对称多处理机系统、大规模并行处理机系统、机群系统和分布式共享存储系统等当代主流并行计算机的结构和关键技术;最后,对并行计算的性能评测和优化进行分析和讨论。

本书第1、6章由王莹编写,第2章由夏又新编写,第4章由陈利编写,第3、5、7章由尹朝庆编写。尹朝庆任主编,负责审定全书的内容和统稿。

作者在编写本书时,曾借鉴或引用了许多专家、学者的著作,在此向他们深表谢意。作者十分感谢华中科技大学出版社对本书出版的指导和支持。

为了便于教师使用本书进行教学,作者制作了本书的PPT电子课件,该课件可从华中科技大学出版社网站免费下载,网址是:www.hustp.com

由于作者水平所限,书中难免有不妥之处,请广大读者批评指正,E-mail:yinchaoqing@yahoo.com.cn。

作 者

2006年7月

目 录

第 1 章 计算机系统结构导论	(1)
1.1 计算机系统结构的基本概念	(1)
1.1.1 计算机系统的层次结构	(1)
1.1.2 计算机系统并行性的基本概念	(5)
1.2 计算机性能评测基础	(10)
1.2.1 计算机性能评测的作用与分级	(10)
1.2.2 计算机的基本性能指标	(12)
1.3 并行计算机的发展	(17)
1.3.1 支持并行计算机的技术进展	(18)
1.3.2 并行结构的发展	(19)
习题一	(22)
第 2 章 指令系统	(24)
2.1 数据表示	(24)
2.1.1 浮点数据表示	(24)
2.1.2 自定义数据表示	(28)
2.2 指令格式优化设计方法	(29)
2.2.1 操作码优化设计方法与评价	(30)
2.2.2 寻址技术	(33)
2.2.3 指令字格式优化设计的措施	(35)
2.3 精简指令集计算机	(37)
习题二	(41)
第 3 章 流水技术与流水处理机	(44)
3.1 指令重叠与先行控制	(44)
3.1.1 指令的顺序执行方式与重叠执行方式	(44)
3.1.2 先行控制技术	(45)
3.2 流水线的分类与时空图	(47)
3.2.1 流水线的分类	(47)
3.2.2 流水线时空图	(48)
3.3 线性流水线的性能计算	(51)
3.3.1 吞吐率	(51)
3.3.2 加速比	(54)
3.3.3 效率	(54)
3.4 非线性流水线的调度与性能计算	(58)
3.4.1 非线性流水线的最优调度方法与性能计算	(58)
3.4.2 非线性流水线的时空图	(64)

3.4.3 非线性流水线动态调度的实现	(65)
3.5 流水线的相关问题与相关处理方法	(66)
3.5.1 局部相关及处理方法	(66)
3.5.2 全局相关及处理方法	(68)
3.5.3 相关对流水线性能的影响	(71)
3.6 多发射处理机及其性能	(74)
3.6.1 超标量处理机及其性能	(74)
3.6.2 超流水处理机及其性能	(77)
3.6.3 超标量超流水处理机及其性能	(77)
3.6.4 超长指令字处理机的基本结构与特征	(78)
3.6.5 多发射处理机的性能比较	(81)
3.7 向量流水处理机及其性能	(83)
3.7.1 向量处理方式	(83)
3.7.2 向量流水处理机的结构	(85)
3.7.3 向量流水处理机的性能	(87)
习题三	(89)
第4章 存储系统	(94)
4.1 存储系统的层次结构与性能指标	(94)
4.1.1 存储系统的层次结构	(94)
4.1.2 存储系统的性能指标	(96)
4.2 并行存储器	(98)
4.2.1 单体多字并行存储器	(98)
4.2.2 低位交叉编址多体并行存储器	(98)
4.3 虚拟存储器	(100)
4.3.1 虚拟存储器的地址变换	(101)
4.3.2 页面替换算法与命中率的计算	(107)
4.3.3 堆栈型替换算法及其堆栈处理过程	(109)
4.4 高速缓冲存储器	(114)
4.4.1 高速缓存的地址映像与地址变换	(114)
4.4.2 高速缓存的替换算法及其实现	(122)
4.4.3 高速缓存的性能分析	(124)
4.5 三级存储系统	(128)
习题四	(129)
第5章 互连网络	(133)
5.1 互连函数	(133)
5.1.1 互连函数的表示方法	(133)
5.1.2 几种基本的互连函数	(134)

5.2 互连网络的结构参数与性能指标	(141)
5.2.1 互连网络的结构参数	(141)
5.2.2 互连网络的性能指标	(142)
5.3 静态互连网络	(143)
5.4 动态互连网络	(149)
5.4.1 多处理机总线	(149)
5.4.2 交叉开关	(151)
5.4.3 多级互连网络	(153)
5.4.4 动态互连网络的比较	(168)
5.5 互连网络的消息传递	(169)
5.5.1 消息格式与消息传递方式	(169)
5.5.2 寻径方法与多播通信	(171)
习题五	(174)
第 6 章 单指令流多数据流计算机	(177)
6.1 单指令流多数据流计算机的基本结构与特点	(177)
6.1.1 单指令流多数据流计算机的两种基本结构	(177)
6.1.2 单指令流多数据流计算机的主要特点	(180)
6.2 分布式存储器 SIMD 计算机实例分析	(181)
6.3 集中式共享存储器 SIMD 计算机实例分析	(185)
6.4 阵列处理机的算法及性能分析	(189)
6.4.1 阵列处理机的差分计算	(189)
6.4.2 阵列处理机的常用算法及性能分析	(190)
习题六	(195)
第 7 章 多指令流多数据流计算机	(197)
7.1 MIMD 计算机结构模型与访存模型	(197)
7.1.1 MIMD 计算机结构模型	(197)
7.1.2 MIMD 计算机访存模型	(200)
7.2 对称多处理机	(204)
7.2.1 对称多处理机的结构	(204)
7.2.2 高速缓存一致性问题	(206)
7.2.3 侦听一致性协议	(208)
7.2.4 SMP 实例分析	(216)
7.3 大规模并行处理机	(218)
7.3.1 大规模并行处理机的结构	(218)
7.3.2 基于目录的一致性协议	(220)
7.3.3 MPP 实例分析	(224)
7.4 机群系统	(229)

7.4.1 机群系统的结构	(229)
7.4.2 机群系统的单一系统映像	(231)
7.5 分布式共享存储系统	(234)
7.5.1 分布式共享存储系统的发展	(234)
7.5.2 共享虚拟存储系统	(235)
7.6 并行计算的性能分析	(240)
7.6.1 并行计算的性能指标	(240)
7.6.2 粒度对并行计算性能的影响	(249)
习题七	(253)
参考文献	(256)

第 1 章

计算机系统结构导论

计算机系统性能之所以不断提高,主要是因为器件的不断更新和计算机系统结构的不断发展。本章主要介绍计算机系统结构的基本概念、系统结构的发展和计算机的基本性能指标。

1.1 计算机系统结构的基本概念

回顾计算机的发展历史,可以看出,计算机系统性能的不断提高主要靠器件的变革和计算机系统结构的改进。器件的迅速发展,使得计算机硬件在体积、重量、速度、可靠性、稳定性等性能上有了极大的改善,而且价格不断降低。可以说,器件技术的突飞猛进,为计算机的发展提供了必不可少的物质基础,是推动计算机发展的重要因素之一。器件的换代是计算机换代的突出标志。

但是,还应该看到,仅有器件的发展是远远不够的。人们还必须研究如何最合理地组织这些器件,如何最大限度地发挥这些器件的作用,如何构成综合性能最佳的系统,这就是计算机系统结构要研究的问题。恩斯洛(P. H. Enslow)曾经比较了 1965 年至 1975 年的器件延迟时间和计算机指令执行时间之间的关系。结果表明,这 10 年间,器件的更新使器件延迟时间降低至原来的十分之一,但计算机指令执行时间却降低至原来的百分之一。由此可见,在这 10 年中,计算机性能提高的幅度比器件性能提高的幅度要大得多。这种情况在近几年的计算机发展中更为明显。除器件的发展以外,计算机系统结构的发展也是推动计算机发展的一个重要因素。因此,计算机换代的标志主要有两个,第一是计算机的器件,第二是计算机系统的结构。

推动计算机系统结构发展的关键是提高计算机系统的并行性。计算机系统结构从低级向高级发展的过程也就是计算机系统的并行性不断发展的过程。

1.1.1 计算机系统的层次结构

在计算机系统中使用的计算机语言可以分成一系列的层次(Level)或级。最低层语言的功能最简单,最高层语言的功能最强,更便于应用。从计算机语言的角度,可将通用计算机系统划分成多级层次结构,每一层以一种不同的语言为特征。按由低层到高层的顺序,各层分别是:微程序机器级、机器语言机器级、操作系统机器级、汇编语言机器级、高级语言机器级和应用语言机器级等。计算机系统的层次结构如图 1.1 所示。每一层的使用者,都可以把该层机器级看成是一台独立的机器,都可以应用相应的机器级语言。

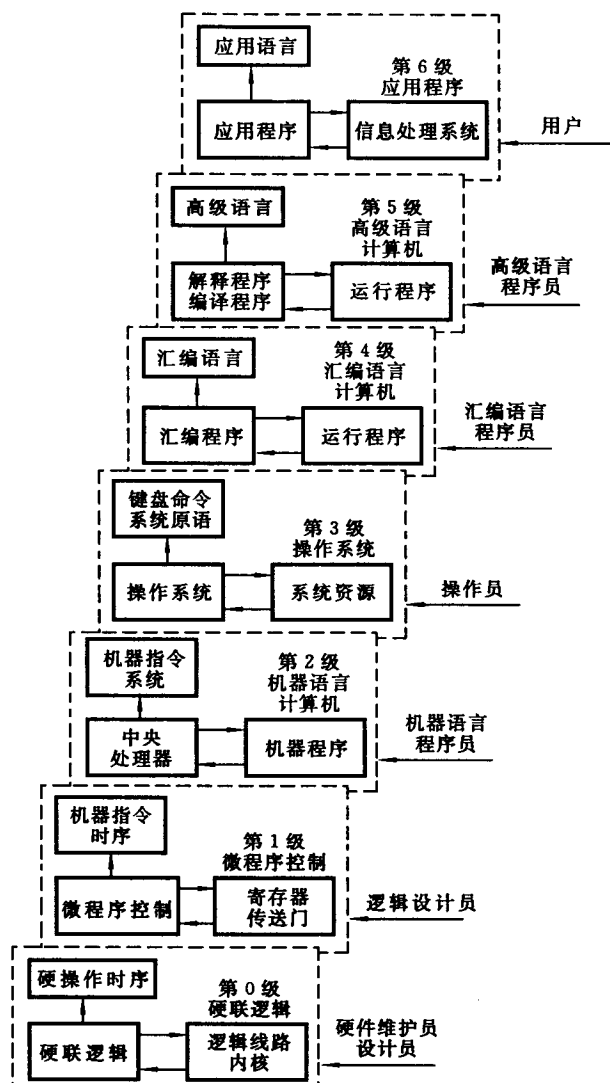


图 1.1 计算机系统层次结构

第0级和第1级是实现机器指定功能的中央控制部分。根据各种机器指令操作所需要的控制时序，配备一套微指令，编写出微程序，来控制信息在各寄存器之间的传送，这就是第1级机器。实现这些微指令本身的控制时序只需要很少的逻辑线路，可采用硬联逻辑实现，它就是第0级机器，是机器的硬件内核。

第2级是机器语言机器。这一级的机器语言是该机的指令系统。机器语言程序员使用指令系统编写的程序由第1级的微程序进行解释。

第3级是操作系统机器。这一级的机器语言中的多数指令是机器语言机器提供的指令，如算术运算、逻辑运算和传送等指令。此外，这一级还提供操作系统级指令，例如，打开文件、读/写文件、关闭文件等指令。用这一级语言编写的程序中的那些与第2级指令相同的指令由微程序解释，而其中的操作系统级指令则由操作系统进行解释。操作系统是运行在第2级机

器上的解释程序。

第4级是汇编语言机器。这一级的机器语言是汇编语言。用汇编语言编写的程序首先被翻译成第3级或第2级语言,然后再由相应的机器进行解释。完成汇编语言程序翻译的程序称为汇编程序。

第5级是高级语言机器。这一级的机器语言就是各种高级语言。用高级语言编写的程序一般由编译程序翻译成第4级或第3级机器上的语言。个别的高级语言也用解释的方法实现其功能。

第6级是应用语言机器。这一级的机器语言是应用语言,是面对非计算机专业人员直接使用的语言。只需在用户终端用键盘或其他方式发出服务请求,就能进入第6级的信息处理系统。

微程序机器级和用组合逻辑控制的机器语言机器级都是用硬件实现的,而采用微程序控制的机器语言机器级是用固件实现的。固件(Firmware)是一种具有软件功能的硬件。以硬件或固件为主实现的机器称为实际机器。

当前,应用语言机器级、高级语言机器级、汇编语言机器级和操作系统机器级都是以软件为主实现的,但也可根据用户对计算机性能/价格(性能价格比)的不同要求增加硬件支持。以软件为主实现的机器称为虚拟机器(Virtual Machine)。

1. 低层机器级对高层机器级的支持

越是高层的机器级语言,其功能越强大,用户使用起来也越方便。但是,各层机器级语言的功能都是依靠下一层机器级的支持才能实现的。而且,这种支持要满足透明性要求。

低层机器级为高层机器级的功能实现提供支持,实现的技术是翻译和解释。翻译(Translate)是先用翻译程序(Translator)把高层机器级上的程序翻译成低层机器级上的等效程序,然后在低层机器级上实现程序功能。解释(Interpretation)是用低层机器级的一些语句或指令来仿真高层机器级上的一条语句或指令的功能,高层机器级程序中的每条语句或指令被逐条解释来实现程序功能。

采用翻译技术实现的典型例子有:用翻译程序将应用语言机器级上的应用程序翻译成高级语言程序,用编译程序将高级语言源程序转换成机器语言目标程序,用汇编程序将汇编语言源程序转换成机器语言目标程序等。采用解释技术实现的典型例子有:用解释方式在机器语言机器级上执行高级语言程序,用微指令程序解释实现机器指令,用微程序或机器指令程序解释实现操作系统的操作命令等。

从计算机系统的某一层的使用者角度看,只需通过该层的语言就可以使用机器,而不必关心其下层的机器级是如何工作和如何实现各自的功能的,这一性质称为透明性(Transparency)。

计算机系统中的“透明”与通常意义下的“透明”在含义上正好相反。通常意义下的“透明”是指公开的,什么都能看得到的,例如,玻璃是透明的,那么,可以看得到玻璃瓶内的任何物体。计算机系统的“透明”是看不到的意思,即对某一层的使用者来说,他看不到该层以下各层的机器属性。

2. 计算机系统结构、计算机组成与实现

(1) 计算机系统结构

计算机系统结构(Computer Architecture)也称为计算机体系结构。经典的计算机系统结构的定义是:计算机系统多级层次结构中机器语言机器级的结构,是软件和硬件/固件的主要界面,是由机器语言程序、汇编语言源程序和高级语言源程序翻译生成的机器语言目标程序能在机器上正确运行所应具有界面结构和功能。

目前,经常使用的是广义的计算机系统结构的概念,它除了包括经典的计算机系统结构的概念范畴外,还包括计算机组成和实现技术。

(2) 计算机组成与实现

计算机组成(Computer Organization)是计算机系统结构的逻辑实现,它主要研究硬件系统在逻辑上是如何组织的,机器级内部数据流和控制流的组成与逻辑设计。计算机实现(Computer Implementation)是指计算机组成的物理实现,主要着眼于器件技术和微组装技术。

(3) 结构、组成与实现之间的相互影响

相同结构的计算机可以因速度等性能要求不同而采用不同的组成,相同的组成可有多种不同的实现方法,这都取决于计算机系统的性能和价格要求以及器件技术的发展情况。

结构不同,其可用的组成技术会有所不同,而不同的组成又会反过来影响系统结构的设计。因此,系统结构的设计不仅需要考虑应用需求,要能为软件和算法的实现提供更多更好的硬件支持,而且要考虑可能采用和准备采用哪些组成技术,不能过多或不合理地限制各种可能的组成与实现技术的采用。

组成与实现可以折中权衡,主要取决于器件的来源和性能价格比。应该在可提供的器件及其实现技术的条件下,在使价格不增或少增的情况下尽可能提高系统的性能。

3. 软件可移植性

软件可移植性(Software Portability)是指在一台机器上编制的软件不用修改或只经少量修改就可在另一台机器上运行,使得同一个软件可以应用于不同的硬件环境中的性能。这样,在过去的计算机系统上使用的大量成熟可靠的软件,特别是应用软件,就可以在新机器上使用而不必重新编写。

在计算机系统结构的范畴内,实现软件可移植性的基本途径有:统一高级语言,采用系列机,以及模拟和仿真等。

(1) 统一高级语言

统一高级语言是指一种对各种应用领域都比较高效的通用高级语言,在结构相同甚至完全不同的机器之间,配备不同的语言翻译程序就可以实现高级语言应用软件的移植。但是,不同的应用对高级语言的语法和语义结构的要求差别较大,因而难以实现一种统一通用的高级语言。从长远的目标来看,还是希望统一高级语言,但是,在目前相当一段时间内,只能是相对的统一。

(2) 系列机

系列机是指具有相同的系统结构,但采用不同组成和实现技术的一系列不同型号的机器。

它们采用不同的组成和实现技术,从而具有不同的性能和价格,但从机器语言机器级的程序设计者角度来看,机器属性是相同的。

系列机技术既可以使同一系列的机器在汇编语言上实现统一,又能在同一系列内开发出性能价格比更高的新机器。在结构相同或相近的机器之间能实现汇编语言应用软件和部分系统软件的移植。

软件兼容(Software Compatibility)是指同一个软件可以不加修改地运行于结构相同的各档机器上,而且运行结果一致。软件兼容有向上兼容、向下兼容、向前兼容、向后兼容之分。向上(下)兼容是指在某档机器上编制的程序可不加修改地运行于比它高(低)档的机器上;向前(后)兼容是指在某型号机器上编制的程序可不加修改地运行于在它之前(后)投入市场的机器上。对于系列机,必须保证做到向后兼容,力争做到向上兼容。

软件向后兼容是系列机的根本特征,也就是说,在先生产出的机器上编制的程序可以不加修改地在后生产的新机器上运行,而后生产的新机器增强了功能和提高了速度,从而提高了性能。因此,对于那些不属于系统结构设计而属于计算机组成和实现的内容,不管是增加、删除,还是修改,都不会影响汇编语言程序和机器语言程序在系列机上的向后兼容。但是,对于属于系统结构设计范畴内的内容,为保证软件向后兼容,只能增加其新的功能或部件,而不能删除或更改已有的功能或部件。例如,可以把CPU与主存之间的数据通路的宽度由16位扩展到32位,以加速主机内部信息的传送;也可以把单总线改为双总线,以减少公共总线的使用冲突。这些内容都属于计算机组成的范畴,在逻辑设计时就要对它进行设计。例如,指令的功能设计与指令字格式设计和编码属于系统设计设计的范畴,如果将原来全部采用定长操作码的指令改成扩展编码操作码,虽然可以减小平均指令字长,但会直接导致以前编写的程序不能正确运行的后果。

不同厂家生产的具有相同系统结构的计算机称为兼容机(Compatible Machine)。

(3) 模拟与仿真

模拟与仿真能实现在结构不同的机器之间的机器语言程序的移植。模拟(Simulation)是用一种机器语言程序解释另一机器的机器指令来实现软件移植的方法。在两种机器的机器指令系统差异较大时,使用模拟方法会使程序运行速度严重下降。仿真(Emulation)是用微程序直接解释另一机器的机器指令来实现软件移植的方法。采用仿真方法可以提高被移植软件的运行速度,但在机器结构差异较大情况下,难以实现仿真。因此,在不同系列机器之间实现软件移植时,可将模拟和仿真两种技术结合起来使用,对于频繁使用且容易仿真的这部分机器指令采用仿真的方法,以提高被移植软件的运行速度;对于较少使用、对速度要求不高且难以仿真的这部分指令及I/O操作,则采用模拟的方法。

模拟与仿真的区别是:模拟是用机器语言程序解释指令,其解释程序存储在主存中;仿真是用微程序解释指令,其解释程序存储在控制存储器中。

1.1.2 计算机系统并行性的基本概念

并行性(Parallelism)是指同一时刻或同一时间间隔内发生两种或两种以上性质相同或不

相同的事件。只要时间上相互重叠,就存在并行性。实际上,并行性包含同时性与并发性两层含义。同时性(Simultaneity)是指两个或多个事件在同一时刻发生。并发性(Concurrency)是指两个或多个事件在同一时间间隔内发生。

1. 发展计算机系统并行性的技术途径

可以通过 3 类技术途径来提高计算机系统的并行性,这就是时间重叠、资源重复和资源共享。

(1) 时间重叠

时间重叠(Time Inteleaving)是在并行性概念中引入时间因素,让多个处理过程在时间上错开,轮流重叠地使用同一套硬件设备的各个部分,以缩短多个处理过程的处理时间的技术。

对指令的各个操作步骤采用流水工作方式是时间重叠的典型例子。如图 1.2 所示,若一条指令的解释可分为取指、分析、执行 3 个操作步骤,并分别使用各自独立的硬件完成,设每个操作步骤完成所需时间皆为 Δt ,那么,第 k 条指令、第 $k+1$ 条指令和第 $k+2$ 条指令就可以在时间上重叠起来,3 条指令彼此在时间上错开 Δt ,以流水方式被解释执行。如果这 3 条指令串行执行,那么需要 $9\Delta t$ 。如果按流水方式执行,如图 1.2 所示,则只需要 $5\Delta t$ 。

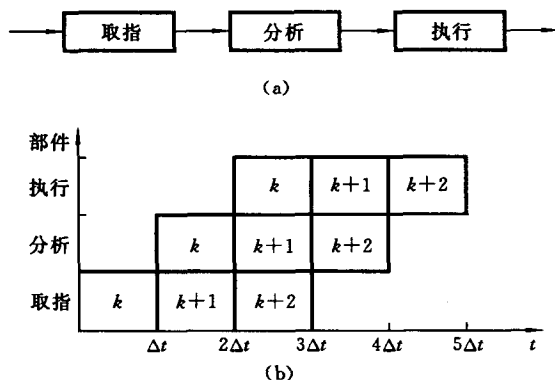


图 1.2 时间重叠的例子

(a) 指令流水线

(b) 指令在流水线各部件中流过的时间关系

由此可见,时间重叠虽没有缩短一条指令的执行时间,但加快了程序的执行速度。时间重叠原则上不需要增加过多的硬件设备就可以提高计算机系统的性能价格比。时间重叠的典型例子有标量流水处理机、超流水处理机和向量流水处理机等。

(2) 资源重复

资源重复(Resource Replication)是在并行性概念中引入空间因素,通过重复设置硬件资源来提高系统可靠性或性能的技术。

如图 1.3 所示,设置 N 个完全相同的处理器(PE),让它们受同一个控制器(CU)控制,控制器每执行一条指令就可以同时让各个处理器对各自分配到的数据完成同一种运算。这是一个利用资源重复提高速度的典型例子。

早期由于受到硬件价格的限制,资源重复以提高可靠性为主,例如,双工系统就是利用资

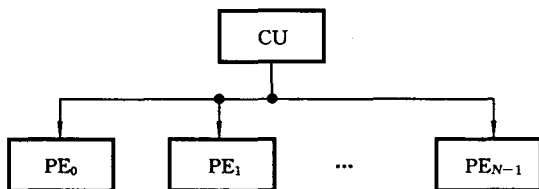


图 1.3 资源重复的例子

源重复,使用两台完全相同的计算机完成同样的任务来提高可靠性的。现在,利用资源重复是为了提高系统的速度性能。这样的典型例子有超标量处理机、阵列处理机等。

时间重叠是实现并行性中的并发性,资源重复是实现并行性中的同时性。

(3) 资源共享

资源共享(Resource Sharing)是利用软件方法让多个任务按一定顺序轮流使用一套资源,通过提高系统资源利用率来提高系统的性能和效率的技术。典型的例子有,多道程序分时系统、计算机网络和分布处理系统等。当然,资源共享不仅仅限于共享系统的硬件资源,也包括软件和信息资源的共享。

2. 并行性等级

并行性可以有不同的等级,从不同的角度来看,并行性等级的划分也不相同。

(1) 按计算机系统中执行程序的并行性划分

并行性等级由低到高可分为4级,它们是:

- ① 指令内部并行,指一条指令内部各操作之间的并行执行;
- ② 指令之间并行,指多条指令的并行执行;
- ③ 任务或进程之间并行,指多个任务或程序段的并行执行;
- ④ 作业或程序之间并行,指多个作业或多道程序的并行执行。

作业或程序级并行的关键是怎样将有限的硬、软件资源有效地同时分配给正在用于求解一个大题目的多个程序。任务或进程级并行主要涉及如何进行任务分解的问题。指令之间的并行主要要解决和处理指令之间存在的相关问题。指令内部的并行主要取决于硬件及组成的设计。总之,并行性等级由高到低反映了硬件实现的比例在增大,所以,并行性的实现也是一个软、硬功能分配问题。随着硬件成本的下降,硬件实现的比例逐步增大。过去在单处理机中并行性升到任务、作业级时,并行性主要是通过操作系统中的进程管理、作业管理,以及并行语言和并发程序设计等软的方法实现。现在在多处理机系统中,由于已有完成各个任务或作业的处理机,因而并行性更多地靠硬件来实现。

(2) 按计算机信息加工的并行性划分

① 存储器操作并行。存储器操作可以采用单体多字、多体单字或多体多字方式在一个存储周期内访问多个字,或者采用位片串字并或全并行方式,在一个存储周期内实现对存储器中大量存储字的高速并行比较、检索、更新、变换等操作。典型的例子是并行存储器和以相联存储器为核心构成的相联处理机。

② 处理器操作步骤并行。处理器操作步骤可以是一条指令的取指、分析、执行等操作步